***Вариант 1***

Написать скрипт на языке оболочки, указанном преподавателем. Скрипт должен принимать как минимум 2 позиционных параметра:

* режим выборки, определяет, какие именно файлы нужно выбрать из текущего каталога, возможные значения:
  + all – все файлы;
  + byname – все файлы, чье имя (без расширения) точно соответствует третьему позиционному параметру;
  + byext – все файлы, чье расширение точно соответствует третьему позиционному параметру;
  + bynsym – все файлы, N-й символ в полном имени (с расширением) которых равен третьему позиционному параметру (где N – четвертый позиционный параметр);
  + bynx – все файлы, полное имя (с расширением) которых отличается от строки, переданной третьим позиционным параметром, одним символом (он может быть любым другим). При этом позиция отличающегося символа в полном имени (точка между именем и расширением не считается) – четвертый позиционный параметр;
* поле информации, определяет, какую информацию о выбранных файлах нужно вывести в файл result.txt в текущем каталоге:
  + name – названия файлов;
  + content – содержимое файлов.

Написать скрипт демонстрирующий работу первого:

* Очищает текущую папку
* Копирует файлы из одной из заранее подготовленных папок с демонстрационным набором файлов – в текущую папку
* Показывает содержимое текущей папки
* Запускает первый скрипт с параметрами:
  + all content
  + byname name test
  + byext content txt
  + bynsym name a 2
  + bynx name tast.txt 2
* Делает 10-секундную паузу между каждым запуском, выводя results.txt

***Вариант 2***

Написать скрипт на языке оболочки, указанном преподавателем. Скрипт должен принимать как минимум 3 позиционных параметра:

* часть списка, определяет, какую часть списка содержимого текущего каталога нужно выбрать:
  + head – вершину (первые N элементов, где N – второй позиционный параметр);
  + tail – хвост (последние N элементов, где N – второй позиционный параметр);
* размер части списка, определяет, сколько элементов из списка попадает в вершину или хвост;
* поле сортировки, определяет, по какому признаку сортировать список содержимого текущего каталога:
  + по имени (по алфавиту)
  + по размеру (начиная с минимального)
  + по расширению (по алфавиту)
  + по дате и времени создания (начиная с самого старого)
  + по дате и времени последнего использования (начиная с самого старого)
  + по дате и времени последнего изменения (начиная с самого старого)
  + по длине полного имени (начиная с самого короткого)
* необязательный четвертый параметр:
  + onlyd – за содержимое текущего каталога считаем только папки;
  + onlyf – за содержимое текущего каталога считаем только файлы.

Таким образом, скрипт получает содержимое текущего каталога, сортирует по определенному признаку, а затем пишет имена выбранной части (выбранного размера) списка содержимого текущего каталога в файл result.txt в текущем каталоге.

Написать скрипт демонстрирующий работу первого:

* Очищает текущую папку
* Копирует файлы из одной из заранее подготовленных папок с демонстрационным набором файлов – в текущую папку
* Показывает содержимое текущей папки
* Запускает первый скрипт с параметрами:
  + head 5 name onlyd
  + tail 3 size onlyf
  + head 7 ext onlyf
  + tail 3 create onlyd
  + head 5 length
* Делает 10-секундную паузу между каждым запуском, выводя results.txt

***Вариант 3***

Написать скрипт на языке оболочки, указанном преподавателем. Скрипт должен принимать как минимум 2 позиционных параметра:

* режим, определяет 1 из режимов скрипта:
  + log – выводит в файл result.txt в текущем каталоге все переменные окружения OS в формате имя=значение (по 1 на строку), отсортированные по полю, указанному во втором параметре по признаку указанному в третьем параметре;
  + triangle – выводит в results.txt текстовый треугольник, состоящий из символа, указанного во втором параметре, и числа строк, указанного в третьем параметре; если третий параметр отсутствует – использовать в качестве него номер запуска скрипта;
* поле сортировки или символ; возможные значения поля сортировки:
  + name – по имени переменной;
  + value – по значению переменной;
* признак сортировки или число; возможные значения признака сортировки:
  + alpha – по алфавиту;
  + length – по длине.

Пример текстового треугольника из 4 строк и символа b:

b

bb

bbb

bbbb

Написать скрипт демонстрирующий работу первого:

* Запускает первый скрипт с параметрами:
  + log name length
  + log value alpha
  + triangle x 3
  + triangle v 8
* Делает 10-секундную паузу между каждым запуском, выводя results.txt

***Вариант 4***

Написать скрипт на языке оболчки, указанном преподавателем. Скрипт должен принимать как минимум 3 позиционных параметра:

* режим, определяет 1 из режимов скрипта:
  + repl – заменяет в именах или содержимом (в зависимости от значения второго позиционного параметра) файлов все символы, указанные в третьем позиционном параметре, на символ, указанный в четвертом позиционном параметре;
  + rem – удаляет в именах или содержимом (в зависимости от значения второго позиционного параметра) файлов все символы, указанные в третьем позиционном параметре;
* поле под замену, определяет, что именно в файлах будет заменяться/удаляться:
  + name – символы в полных именах (с разрешением);
  + content – символы в содержимом;
* символ под замену/удаление, определяет, какой символ будет заменяться/удаляться;
* новый символ, определяет, на что будут заменяться исходные символы.

Написать скрипт демонстрирующий работу первого:

* Очищает текущую папку
* Копирует файлы из одной из заранее подготовленных папок с демонстрационным набором файлов – в текущую папку
* Показывает содержимое текущей папки
* Запускает первый скрипт с параметрами:
  + repl name a b
  + repl content 2 3
  + rem name 1
  + rem content x
* Делает 10-секундную паузу между каждым запуском, выводя содержимое текущей папки

***Вариант 5***

Написать скрипт на языке оболчки, указанном преподавателем. Скрипт должен принимать как минимум 2 позиционных параметра:

* оператор, определяет, какая операция будет применяться к массиву операндов:
  + add – складывает все операнды (1+2+3+4+5) между собой и пишет сумму в result.txt в текущей папке;
  + sub – вычитает поочередно все операнды из первого (1-2-3-4-5) и пишет результат в result.txt;
  + multi – умножает друг на друга все операнды (1\*2\*3\*4\*5) и пишет произведение в result.txt;
  + div – делит первый операнд поочередно на все остальные ((((1/2)/3)/4)/5) и пишет результат в result.txt;
  + fact – вычисляет факториал всех переданных операндов и пишет результаты (по значению на строку) в result.txt;
  + pow – возводит каждый операнд в степень, переданную третьим позиционным параметром и пишет результаты (по значению на строку) в result.txt;
* массив операндов – строка вида ‘1 2 3 4 5’, где 1, 2, 3, 4 и 5 – операнды;
* степень для режима pow.

Написать скрипт демонстрирующий работу первого:

* Запускает первый скрипт с параметрами:
  + add ‘4 8 15 16 23 42’
  + sub ‘4 8 15 16 23 42’
  + multi ‘4 8 15 16 23 42’
  + div ‘4 8 15 16 23 42’
  + fact ‘4 8 15 16 23 42’
  + pow ‘4 8 15 16 23 42’ 2
* Делает 10-секундную паузу между каждым запуском, выводя содержимое results.txt